

PATENT APPLICATION

NEURAL NETWORK FOR AEROELASTIC ANALYSIS

Inventor(s): Darin P. Haudrich, a citizen of The United States, residing at
7671 State Rt. 156
Waterloo, IL 62298

Dale M. Pitt, a citizen of The United States, residing at
6145 Southern Drive
Affton, MO 63123

Assignee: The Boeing Company
100N. Riverside
Chicago, IL 60606-1596

Entity: Large Business Concern

NEURAL NETWORK FOR AEROELASTIC ANALYSIS

BACKGROUND OF THE DISCLOSURE

Field of the Disclosure

5 [0001] The disclosure relates generally to the field of aeroelastic analysis. More particularly, the present disclosure relates to a neural network for aeroelastic analysis.

Description of Related Art

[0002] Aeroelastic analysis is typically used in the aerospace industry to characterize aircraft and structures that form part of or that affect the stability of the aircraft. Aircraft are increasingly
10 designed for and subjected to longer service lives. As a result, one or more structures on an aircraft may be repaired or altered during the life of the structure. Any repair or alteration of an aircraft structure may affect the aeroelastic characteristics of the aircraft.

[0003] To ensure that a structural repair or modification does not detrimentally affect the safety or flight characteristics of the aircraft, the repaired or modified structure may be subjected
15 to aeroelastic analysis, such as aeroelastic stability or flutter analysis.

[0004] A flutter assessment for repairs of a surface, such as a surface of a wing or tail section, can be more detailed than non-repair flutter analysis because the analysis typically needs to determine the effects of additional mass. Each individual repair to a surface may need to be accounted for in the analysis, and there may be numerous surface repairs. The desire to perform
20 aeroelastic analysis may be heightened when the locations and weights of repairs exceed a predetermined category of approved repair parameters.

[0005] Typically, technicians at a repair facility provide the exact locations and weights of the repairs to an engineering team that includes an aeroelastic engineer. The aeroelastic engineer can conduct a detailed structural analysis using a finite element model of the structure. It may
25 take several hours to days to perform the structural analysis for the finite element model. The aeroelastic engineer uses the results of the structural analysis to ensure the modifications or repairs do not affect the structure in a manner that changes critical flutter mechanisms of the structure.

[0006] The aeroelastic engineer uses the results of the finite element model to perform flutter analysis. The aeroelastic engineer may, for example, create mode shapes for the model for one or more of the modes critical to flutter mechanism. The entire process from initial repair to return of flutter assessment may take on the order of one to two weeks.

5 [0007] Because the safety of the aircraft may be in question after performance of the repair, the time required to perform the aeroelastic analysis may represent a period in which the aircraft is grounded and unavailable. The process of performing the aeroelastic analysis is costly and having downed flight time for aircraft is exceedingly costly.

10 BRIEF SUMMARY OF THE DISCLOSURE

[0008] A system and method of performing aeroelastic analysis using a neural network are disclosed. Input parameters, such as mass and location, contributing to aeroelastic stability characterization are determined and constrained. A model of a structure to be analyzed can be constructed. The model can include a number of locations where the input parameters can be
15 varied. The aeroelastic characteristic of the structure can be analyzed using a finite element model to determine a number of output characteristics, each of which can correspond to at least one of a plurality of input samples. A neural network can be generated for determining the aeroelastic characteristic based on input parameters. The input sample/output characteristic pairs can be used to train the neural network. The weights and bias values from the trained neural
20 network can be used to generate a non-linear transfer function that generates the aeroelastic characteristic in response to input parameters.

[0009] The features, functions, and advantages can be achieved independently in various embodiments of the present disclosure or may be combined in yet other embodiments.

25 BRIEF DESCRIPTION OF THE DRAWINGS

[0010] The features, objects, and advantages of embodiments of the disclosure will become more apparent from the detailed description set forth below when taken in conjunction with the drawings, in which like elements bear like reference numerals.

[0011] Figure 1 is a functional block diagram showing parameter relationships in an aeroelastic analysis.

[0012] Figure 2 is an example of a structural model used in aeroelastic analysis.

[0013] Figure 3 is an example of aeroelastic analysis performed on a structural model.

5 [0014] Figures 4A-4B are functional block diagrams of an embodiment of a system for performing aeroelastic analysis according to the disclosure.

[0015] Figure 5 is a functional block diagram of an embodiment of a computer that can be configured to perform aeroelastic analysis in accordance with the disclosure.

[0016] Figures 6A-6C are flowcharts of processes used in performing aeroelastic analysis.

10 [0017] Figure 7 is a graph of example input parameters generated using a random process.

[0018] Figure 8 is a screen image of an embodiment of an aeroelastic analysis user interface.

[0019] Figure 9 is an example of a contour plot generated using aeroelastic analysis.

[0020] Figures 10A-10B are graphs of examples comparing aeroelastic analysis performed according to the disclosure against the same analysis performed using initial finite element
15 model structural analysis.

[0021] Figure 11 is a graph of error in calculated and predicted aeroelastic characteristics.

DETAILED DESCRIPTION OF THE DISCLOSURE

[0022] A system and method for performing aeroelastic flutter analysis based in part on a
20 neural network. A structure can be modeled for flutter analysis. The model can resemble a finite element model used for structural analysis. One or more parameters contributing to the flutter characteristics can be defined and constrained. For example, each of the parameters may be constrained to a predetermined range or set. One or more positions or locations on the model can be configured such that one or more of the parameters may be altered at each of the one or more
25 locations.

[0023] A set of input parameters can be determined, with input parameters corresponding to at least a subset of available positions on the model. The set of input parameters are used as inputs

to a flutter analysis. Results of the flutter analysis for each of the input parameters can be determined using, for example, a finite element structure analysis followed by interpretation of the results in a flutter analysis. The input parameters and corresponding flutter analysis outputs can form a set of input/output pairs.

5 **[0024]** An artificial neural network can be defined or other wise constructed. The artificial neural network can include an input layer, one or more neurons in a hidden layer, and an output layer. The neurons in the hidden layer can each include a weight, a bias, and a transform function.

10 **[0025]** The input/output pairs can be used to train the artificial neural network. The neural network can thus be trained to provide an output substantially equal to the output obtained using the finite element model in response to the application of the corresponding input parameters. Thus, application of input parameters corresponding to a specific location in the structure model can result in substantially the same output regardless of whether the finite element model or neural network is used to determine the output.

15 **[0026]** Once the neural network is trained, it can provide general solutions. The neural network can provide accurate results for input parameters that were not part of the original training set. The neural network can converge to a solution based on the initial training set and may not need to further adjust the values in the neurons. Thus, the values in the neurons in the neural network may be used to generate a module or software that performs a transformation of
20 the input parameters to a flutter solution. The module or software need not perform as a neural network but can provide the transform function that is defined by the neural network. The module or software can be configured such that it may be implemented in a device or apparatus that is accessible at the point of repair. The flutter analysis can then be performed in a relatively short period of time after performing the repair or modification. There can be a shortened period
25 of time between completing a repair or modification and receiving the results of the aeroelastic analysis. The shortened period of time can substantially reduce the costly down period of an aircraft.

30 **[0027]** Aeroelastic stability of an aircraft structure can be determined when additional mass is added to repair an aircraft structure. The analysis may be of greater interest when the repair is performed on an aircraft surface. Figure 1 is a prior art functional diagram showing the

relationship between three forces that contribute to the flutter of a structure. The three forces include aerodynamic 110, elastic 120 or stiffness, and inertial 130 that is dependent on mass. The flutter analysis may need to be re-determined for a change in any of the three forces. For example, a repair of a structure can change the inertial force 130 when mass is added during the course of the repair.

[0028] A set of input parameters can be determined based on the knowledge of the force relationship shown in Figure 1. A surface repair of an aircraft structure, such as a wing, tail, or stabilator typically does not affect the aerodynamic forces 110 experienced by the structure. Typically, a repair or modification changes the elastic force 120 or inertial force 130. However, because a repair technician typically does not have the ability to determine changes in elastic force 120, it may not be convenient to define input parameters directed to changes in elastic force 120. Changes in the inertial force 130 can be related to the amount of mass added in the repair and the position of the mass in the structure. Because a repair technician can typically obtain or otherwise discern a mass of the repair and location of the repair, mass and location are typically good candidates for input parameters. Additionally, repair mass and repair locations can be good candidates for input parameters because these are the typical parameters used in a finite element model of the structure.

[0029] Figure 2 is an embodiment of a structural model 200 used in aeroelastic analysis. The structure can be, for example, a stabilator, wing, elevator, canard, aileron, flap, spoiler, stabilizer, tail section, rudder, and the like, or some other structure.

[0030] For the purposes of explanation, the structural model of an aircraft stabilator is shown. The structural model 200 includes a plurality of identified positions, for example 202, which correspond to locations where mass may be added as a result of a repair. Although twenty-four locations are defined in the structural model 200 of Figure 2, any number of locations may be defined. The actual number of locations used to characterize a structure may be determined based in part on the sensitivity of the structure to mass changes at different locations and the desired sensitivity of system analysis. The positions can be used in the flutter analysis of the disclosure, but are not necessary for the conventional aeroelastic analysis.

[0031] In a conventional flutter analysis, a finite element model of the structure is analyzed before and after mass from repairs are added to the structure. In the finite element model, each

individual repair made to a surface of the structure can be accounted for in the analysis. The weights and locations of each of the repairs to a structure can be included to a finite element model. The finite element model can be structurally analyzed. The results of the structural analysis can be examined to ensure that the repair weights do not affect the structure in a manner that changes a critical flutter mechanism. In one process, flutter analysis can be performed by generating mode shapes for the structural model 200 for one of the modes contributing to the flutter mechanism.

[0032] In the conventional flutter analysis using a finite element model, the structural analysis may take several hours to perform. The finite element model of the repaired structure can be compared to the finite element model of the original non-repaired structure to determine if the two structures behave similarly. If it appears that the repair has not adversely affected the structure, a flutter analysis can be performed.

[0033] A conventional manner of analyzing flutter is to generate a series of V-g- ω curves that characterize the structure. On the V-g- ω curves, frequency and damping are plotted as a function of the velocity. Figure 3 is an example of a V-g- ω curve generated using the results of a finite element analysis. Each of the modes operates at a frequency that remains relatively constant over a range of velocities. Each of the various modes contribute a curve on the lower plot that corresponds to a damping curve plotted on the upper plot. The point at which the damping curve crosses over the origin identifies the flutter speed and flutter frequency.

[0034] In the conventional analysis, an engineer performs the finite element analysis and then generates the series of V-g- ω curves. The engineer then interprets the data in the V-g- ω curves to determine the flutter speed of the repaired structure.

[0035] The disclosed aeroelastic analysis system and method does not require the time consuming analysis of the conventional process. However, the disclosed aeroelastic system and method use results obtained from the conventional analysis as training data for a neural network.

[0036] Figure 4A is a functional block diagram of an aeroelastic analysis system 400 that can be configured to perform flutter analysis. The system 400 can include an input module 410 coupled to a neural network module 420 that is, in turn, coupled to an output module 430. A processor

440 coupled to memory 442 can be coupled to the input module 410, neural network module 420 and output module 430.

[0037] The input module 410 can include an Input/Output (I/O) device 412 and a memory 414. The input module 410 can be configured to receive input parameters that are used in the
5 aeroelastic or flutter analysis. For example, the input module 410 can receive, via the I/O device 412, the weight and position information for repairs made to a particular structure.

[0038] The I/O device 412 can be configured to receive the parameters via manual data entry or may receive the parameters via an electronic interface. The I/O module can also be configured to provide an output that confirms receipt of the input parameters. For example, the
10 I/O device 412 can include a display that is configured to echo data that is manually entered to the input module 410.

[0039] The I/O device 412 can include, for example, a keyboard, keypad, computer mouse, trackball, buttons, switches, slides, knobs, dials, and the like, or some other device configured to accept manual entry. Additionally or alternatively, the I/O module 412 can include, for example,
15 an electronic port, an electrical connector, a receiver,, a wireless receiver, an optical reader, an optical detector, a magnetic reader, a magnetic detector, and the like, or some other assembly for receiving an electronic input.

[0040] The input module 410 can receive the input parameters and store them in the associated memory 414. In other embodiments, the input module 410 can store the received input
20 parameters in the centralized memory 442. In still other embodiments, the input module may provide the received input parameters to the neural network module 420 as they are received.

[0041] The neural network module 420 can be configured as a neural network having a predetermined number of neurons and one or more predetermined transfer functions. Configuring the neural network module 420 with a neural network can be particularly
25 advantageous when training the neural network to converge on a solution. In another embodiment, the neural network module 420 is configured to implement a non-linear transfer function implemented by a neural network trained using a predetermined training set of input and output data. In such an embodiment, the neural network module 420 is not configured as a

neural network but instead, performs substantially the transform function of a previously trained neural network.

[0042] An embodiment in which the neural network module 420 is configured to perform substantially the transfer function of a previously trained neural network can be advantageous for implementations of the system 400 in devices having relatively low computational power. A first flutter analysis system 400 can be configured with a neural network. The neural network can be trained with a training set of data. Then the transfer function of the trained neural network can be implemented in the neural network module to provide the transfer function of the neural network without the need to actually implement a neural network.

[0043] However, regardless of whether the neural network module 420 implements a neural network or a transfer function of a trained neural network, it is convenient to model the neural network model in terms of the trained neural network. Thus, the neural network module 420 can include a weight vector module 422 coupled to an input of a summer 424. In one embodiment, the weight vector module 422 can be the same dimension as the input parameters. In another embodiment the weight vector module 422 can be the same dimension as the number of neurons in the neural network. The weight vector module 422 can be configured to multiply the input parameters by a weighting vector to weight the input parameters prior to summing and passing through the transfer function.

[0044] A bias vector module 426 can be coupled to another input to the summer 424. The bias vector module 426 can be configured to provide a vector or a scalar value to the summer. The summed output can be coupled to a transfer function module 428.

[0045] The transfer function module 428 is typically a predetermined nonlinear transfer function, although linear transfer functions may also be used. In one embodiment, the transfer function implemented in the transfer function module 428 is a sigmoid function. In other embodiments, the transfer function can include a tangent sigmoid, a hyperbolic tangent sigmoid, a logarithmic sigmoid, a linear transfer function, a saturated linear transfer function, a radial basis transfer function, and the like, or some other type of transfer function.

[0046] The output module 430 can include a display 432, another output device 434, and can include memory 436. The output device 434 can be an electronic output interface. For example,

the output device 434 can be a electrical connector, a communication device, a communication port, a modem, an optical transmitter, a wireless transmitter, a display driver, and the like, or some other device for outputting results.

[0047] The display 432 can be configured to output the results received from the neural network module 420. In one embodiment, the display 432 can be configured to output a flutter speeds and flutter frequency for a number of damping factors. In another embodiment, the display can be configured to output a graph, list, or plot of aeroelastic analysis data.

[0048] The memory 436 in the output module 430 can be configured to store one or more outputs of the neural network module 420 for output in a single file, list, or graph. For example, the memory 436 can be configured to store a number of data points for plotting on a graph that is shown on the display 432. In other embodiments, the output module 430 can use the shared memory 442.

[0049] The shared memory 442 and processor 440 can be configured to operate in conjunction with the modules. In an embodiment of the system 400, the processor 440 and memory 442 can be configured to perform one or more of the functions of the modules 410, 420, and 430. For example, one or more functions performed by the modules can be configured as processor readable instructions stored in memory 442 and executed by the processor 440. In one embodiment, the neural network function is configured as a number of processor readable instructions stored in memory 442 and executed by the processor 440.

[0050] The memory 442 can be a single storage device or may be multiple storage devices. Similarly, the processor 440 can be a single processor or can be a plurality of processors.

[0051] Figure 4B is a functional block diagram of an embodiment of a neural network 450 that may be implemented in the neural network module 420 of Figure 4A. The neural network 450 includes an input layer 452 coupled to a neuron layer 454. the output of the neuron layer 454 can be coupled to an output layer 456. Although the neural network 450 includes a single neuron layer, other embodiments can include a plurality of neuron layers disposed between the input layer 452 and the output layer 456. Although a feed forward neural network 450 is shown in Figure 4B, the neural network 4B may have any type of architecture and is not limited to a feed forward architecture.

[0052] In the embodiment shown in Figure 4B, the neural network 452 includes an input layer that can be configured to accept a multiple dimensional input signal that includes the weight of a repair and the location of the repair on the modeled structure. The neuron layer 454 can include a number of neurons. Each neuron can have an associated weight vector and bias. The number of neurons in the neuron layer 454 can correspond to the dimension of the output layer 456. Each neuron in the neuron layer 454 can, for example, determine an output value for the flutter analysis. For example, a neuron may be configured to determine a flutter speed or a flutter frequency for a particular damping factor.

[0053] In one embodiment, the neural network 450 can be configured to generate a flutter analysis for an aircraft structure and can be configured to provide the flutter speed and flutter frequency for a number of damping factors. The neural network 450 can comprise a single layer, feed forward neural network. The neurons can each be configured to implement a hyperbolic tangent sigmoid function.

[0054] The network can be trained using a known set of flutter results. The input layer 452 can be configured to receive the input parameters that include the weight and locations of repairs made to the aircraft structure. The neural network can be trained with the known results of the training set until an error in the results of the neural network is less than a predetermined amount. The architecture of the neural network can, through the iterative training process, attempt to minimize a mean square error. The neural network can be characterized as converging on a solution when the error produced by the neural network is less than a predetermined error limit.

[0055] As described earlier, one or more of the modules in the aeroelastic analysis system can be implemented as software executed by a processor. Figure 5 is functional block diagram of an embodiment of a computer 500 that can be configured to perform a portion or all of one or more of the modules in the aeroelastic analysis system. The computer 500 can include a display 510, I/O devices 550 including a keyboard 552 and an input device 554, a processor 520, memory 524, an I/O controller 540, a hard drive 562, one or more removable storage drives 564, which can include a floppy drive, an optical storage 566, some other storage devices 568, a communication device 530 such as a modem, and a network interface card (NIC) 534. The various elements can be coupled using one or more computer busses 502 within the computer 500. The one or more storage devices 568 can include, but are not limited to, ROM, RAM, non-

volatile RAM, flash memory, magnetic storage, optical storage, tape storage, hard disk storage, and the like, or some other form of processor readable media for storing data, instructions, or a combination of data and instructions.

[0056] Figure 6A is a flowchart of an embodiment of a method 600 of performing aeroelastic analysis. The method 600 can be implemented, for example, in designing a system for use in the system 400 of Figure 4A, the computer 500 of Figure 5, or a combination of the two.

[0057] The method 600 begins at block 600 where an engineer or other system designer determines one or more input parameters. For example, in an artificial neural network used to predict flutter speeds and flutter frequencies for a lifting surface that was changed by repairs or modifications, it may be assumed that the stiffness of the surface has not changed. Thus, the mass and locations of the mass may be the parameters that are accounted for in the analysis. The input parameters may be further constrained such that the initial assumption that the change does not affect stiffness remains accurate. In one embodiment, the locations of the repairs or modifications were constrained to the locations illustrated in the structural model of Figure 2.

Additionally, the repair or modification weights were constrained to no greater than 10% of the total weight of the structure. In an example where the structure is an aircraft stabilator weighing three hundred pounds, the weights of the repairs are constrained to not exceed thirty pounds.

[0058] Once the input parameters are defined, the engineer can proceed to block 614 and determines a training set of characteristic pairs. The neural network is trained according to the characteristic I/O pairs. Thus, the engineer can generate one or more input parameters and the resultant outputs for those inputs. Thus, the engineer may generate a number of weights and locations corresponding to hypothetical repairs. The engineer can then determine the flutter analysis results for the input parameters using, for example, a conventional analysis method such as a finite element model and V-g- ω plots.

[0059] After determining the characteristic I/O pairs, the engineer can proceed to block 620 and generate a neural network. For example, the engineer may generate a feed forward neural network where the neurons use a hyperbolic tangent sigmoid as the transfer function.

[0060] After the neural network is generated, the engineer can proceed to block 622 and train the neural network using the characteristic I/O pairs. The neural network can be trained with a

predetermined number of characteristic I/O pairs, or the neural network may be trained until an error in the result generated by the network is below a predetermined error level. For example, the neural network can be trained until the mean square error is below a predetermined error threshold.

5 **[0061]** After training the neural network, the engineer can proceed to block 626 and determine the weight vectors and bias values that are implemented in the converged neural network. The engineer can proceed to block 626 and verify the accuracy of the neural network, for example, by providing input parameters to the neural network for which the results are known. The known results can be compared to the output from the neural network to determine the accuracy of the
10 output.

[0062] The engineer can proceed to block 628 and determine the generality of the solution by providing input parameters to the neural network that were not part of the initial training set. The engineer can compare the expected results against the results output by the neural network. If the neural network demonstrates satisfactory generality, the process may be complete and
15 subsequent inputs may be provided to the neural network and the results obtained from the neural network.

[0063] However, once the neural network is trained, continued use of a neural network is not required. The engineer can proceed to block 680 and generate a simplified module that performs the transfer function characterized by the weight and bias values of the trained neural network.
20 The transfer function of the neural network can be implemented, for example, in a hand held device, or some other type of computing device that can be used by a repair technician to verify the aeroelastic analysis of a repaired structure.

[0064] Figure 6B is a flowchart of a method 640 that can be implemented, for example, by the system 400 of Figure 4A, the computer 500 of Figure 5, or some combination of the two.

25 **[0065]** The method 640 begins at block 642 where the system receives input parameters. The system can receive the input parameters, for example, using an input module that accepts user inputs of weights and locations of the weights. The system can then proceed to block 644 and apply the neural network transfer function to the input parameters to produce an output that is substantially equal to the output that is produced by a trained neural network. The system can,

for example, implement a neural network or can emulate the transfer function of the neural network. The system then proceeds to block 646 and outputs the results. The results can be output, for example, using a display or other output device.

[0066] Figure 6C is a flowchart of a method 644 of applying a neural network transfer function to input parameters, such as described in the method of Figure 6B. The method 644 begins at block 652. The system multiplies the input parameters by the weight vector of the trained neural network. The system then proceeds to block 654 and sums the weighted input parameters. The system then proceeds to block 656 and sums an additional bias value to the sum of the weighted input parameters. In some embodiments, the bias value may be zero. In other embodiments, the bias value is non-zero. After computing the entire sum, the system can proceed to block 658 and apply the sum to the neuron transfer function. Note that the neuron transfer function referred to in block 658 represents the transfer function implemented in a neuron of the neural network and does not refer to the transfer function of the entire neural network referred to in the flowchart of Figure 6B.

[0067] Figure 7 is a graph 700 of example input parameters generated using a random process. The graph 700 shows a number of weights and locations of hypothetical or actual repairs or modifications. The input parameters can be used as the input parameters for the characteristic I/O pairs used to train the neural network. For each input parameter example, a conventional aeroelastic analysis can be performed to generate a result. The input parameter and conventional result are used as a single I/O pair in the set of characteristic I/O pairs used to train the neural network. As seen in the graph 700, at least one training pair is generated corresponding to each location on the structural model.

[0068] The input parameters for the training set can be generated randomly, as shown in the graph of Figure 7, or may be generated to uniformly cover the design space or provide some other distribution over the design space.

[0069] Figure 8 is an embodiment of a screen image of an aeroelastic analysis user interface 800. The user interface 800 can be, for example, implemented as part of the input module 410 and output module 430 of the system shown in Figure 400. The user interface 800 can include a structural model window or display 810 that displays or otherwise provides the structural model, showing the locations and reference identifiers for a number of input parameters. In the user

interface 800 of Figure 8, the structural model window 810 shows a structural model of a stabilator. The structural model includes a number of reference numbers indicating the possible locations for weight to be added to the structure.

5 [0070] An input window 820 or input section to the user interface 800 can include one or more fields configured to accept user input. For example, the input window 820 includes a first field 822 configured to accept a weight value. The input window 820 also includes a second field 824 configured to accept a grid point value corresponding to one of the reference locations identified in the structural model shown in the structural model window 810.

10 [0071] Although the input fields are shown as open fields that are to be filled by a user, the fields can be configured as drop down menus or some other type of entry. A limited entry field may be advantageous for fields that have a limited number of possible values such as, for example, as the grid point input field.

15 [0072] The user interface 800 can also include an output window 830 or output section that can include one or more fields configured to present results to the user. In the embodiment shown in Figure 8, the output window includes fields that provide the flutter speed and flutter frequency for three different damping values. The output values can be determined based in part on the values provided in the input fields.

20 [0073] The embodiment of Figure 8 includes six output fields. A first pair of output fields, 842 and 844, is configured to provide the flutter speed and flutter frequency for a 0.00 damping value. A second pair of output fields, 852 and 854, is configured to provide the flutter speed and flutter frequency for a 0.02 damping value. A third pair of output fields, 862 and 864, is configured to provide the flutter speed and flutter frequency for a 0.04 damping value.

25 [0074] The user interface 800 is not limited to providing numerical output. Figure 9 is an example of a contour plot generated using aeroelastic analysis. The user interface 800 can be configured to provide the contour plot 900 in response to user input. The contour plot is an example of a plot of flutter speeds as a function of weight and pitch inertia. The aeroelastic analysis system can be configured to output other analysis results including, but not limited to, control surface buzz, ejection store loads, flight flutter testing, and the like, or some other analysis.

[0075] Figures 10A-10B are graphs comparing aeroelastic analysis performed according to one embodiment of the disclosure against the same analysis performed using initial finite element model structural analysis. In Figure 10A, the flutter speeds for a 0.02 damping value calculated according to the finite element model plotted and compared against the flutter speeds determined using a neural network based flutter analysis. Figure 10B is graph of the flutter frequency for a 0.02 damping value calculated according to the finite element model plotted and compared against the flutter frequency determined using a neural network based flutter analysis. As can be seen from the two graphs, the flutter frequency determined using the two methods are comparable.

[0076] Figure 11 is a graph of percentage error in flutter speeds calculated according to the finite element method and flutter speeds determined using a neural network based aeroelastic analysis. Over the entire design space, the maximum error exhibited is less than 1%. Because conventional aeroelastic analytical results may differ from actual results by greater than 1%, the level of error may be acceptable.

[0077] An artificial neural network based aeroelastic analysis system and method are thus disclosed. The system allows aeroelastic analysis to be accurately performed in a fraction of the time used for finite element modeling and V-g- ω analysis. Timely aeroelastic analysis can reduce the downed aircraft time and costs typically associated with performing extended aeroelastic analysis. In addition to reducing the time required to perform the actual numerical analysis, the system decreases a time required for an aeroelastician to analyze repairs. The system may be used by the repair technician to analyze aeroelastic effects of the repair, and an engineer would not need to spend any time if the results from the system indicate that the repair has little or substantially no effect on flutter speeds. The system does not replace aeroelastic analysis, but instead provides an additional tool for performing the analysis. The aeroelastic analysis process can be simplified to allow non-aeroelastic engineers to produce accurate flutter speed determinations that can be used to make a flight certification decision.

[0078] An artificial neural network can be generated to perform the aeroelastic analysis. In one embodiment, the artificial neural network can be configured to determine flutter speeds and associated flutter frequencies for aircraft structures. The structure is modeled and the model is used to determine one or more input parameters to be used by the neural network. The artificial

neural network can be trained with a set of training data having one or more I/O characteristic pairs.

5 [0079] The trained neural network, or a system or apparatus that emulates the trained neural network, can be implemented in a system for use by repair personnel. The system can include an input module configured to receive one or more input parameters. The system can also include an output module to present one or more aeroelastic analysis results that are generated, based at least in part, on the input parameters.

10 [0080] The above description of the disclosed embodiments is provided to enable any person skilled in the art to make or use the disclosure. Various modifications to these embodiments will be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other embodiments without departing from the scope of the disclosure. Thus, the disclosure is not intended to be limited to the embodiments shown herein but is to be accorded the widest scope consistent with the principles and novel features disclosed herein.